IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of the claims in the application:

1. (Currently amended) A method of making a computational service available in a multiple server computing environment comprising:

exchanging information between a plurality of servers;

initiating a connection between a client unit and a first server of said plurality of servers;

determining a most recently accessed session of a plurality of sessions on said plurality of servers;

determining at said first server a location of ~~a session~~ said most recently accessed session on one of said plurality of servers; and

redirecting said client unit via said first server to a second server of said plurality of servers having said most recently accessed session;

wherein each of said plurality of sessions comprises a plurality of services requested by said client unit;

wherein said first and second servers can each provide said plurality of services; and

wherein said plurality of services comprise state maintenances for a user of said client unit.

2. (Previously presented) The method of Claim 1, wherein said initiating comprises:

said client unit broadcasting a message to said plurality of servers; and

said first server responding to said message.

3. (Previously presented) The method of Claim 1, wherein said initiating is in response to a prior server failing.

4. (Currently amended) The method of Claim 1, wherein said most recently accessed session is associated with a token.

5. (Currently amended) The method of Claim 4, wherein said second determining step comprises:

said first server sending a message to said plurality of servers, said message comprising said token; and

said plurality of servers responding to said first server with session information associated with said token.

6. (Cancelled)

7. (Previously presented) The method of Claim 1, further comprising securing messages between said client unit and said plurality of servers.

8. (Previously presented) The method of Claim 7, wherein said securing is performed with a keyed hash signature.

Claims 9-16 (Cancelled)

17. (Previously presented) The method of Claim 1, wherein said information exchanged between said plurality of servers comprises a description of a network topology of said plurality of servers.

18. (Previously presented) The method of Claim 17, further comprising updating status in said network topology on a relationship between a connectivity of said client unit and said second server.

19. (Previously presented) The method of Claim 1, wherein said second server comprises a server available for having said session.

20. (Previously presented) The method of Claim 1, wherein said client unit comprises a thin client unit.

21. (Previously presented) The method of Claim 1, wherein said session comprises a thin client session.

22. (Previously presented) The method of Claim 1, comprising:
maintaining said session persistently by said plurality of servers.

23. (Previously presented) The method of Claim 1, wherein said client unit comprises a stateless device.

24. (Previously presented) The method of Claim 1, wherein said determining said location at said first server of said session on one of said plurality of servers comprises receiving a message from said second server of an availability of said second server for having said session.

25. (Currently amended) The method of Claim 24 ~~14~~, wherein said token can identify a plurality of sessions.

26. (New) The method of Claim 1, wherein said plurality of servers communicate with each other in a self-organizing manner.

27. (New) The method of Claim 26, wherein said plurality of servers do not have a master component.

28. (New) A method of making a computational service available in a multiple server computing environment comprising:

exchanging information between a plurality of servers via a self-discovery mechanism;

initiating a connection between a client unit and any one of said plurality of servers that is available to connect with said client unit by inserting a token into said client unit;

finding a plurality of sessions associated with said token;

determining a most recently accessed session;

directing said client unit to a first server of said plurality of servers having said most recently accessed session; and

redirecting said client unit via said first server to a second server of said plurality of servers having a next most recently accessed session if said first server fails.